Managing a Cluster of IoT Brokers in Support of Smart City Applications

Shadha Tabatabai, Ihab Mohammed, and Ala Al-Fuqaha
Computer Science Department
Western Michigan University
Email: {shadhamuhinoo.tabatabai,ihabahmedmoha.mohammed, ala.al-fuqaha}@wmich.edu

Mohammad A. Salahuddin
David R. Cheriton School of Computer Science
University of Waterloo
Email: mohammad.salahuddin@ieee.org

***Abstract*—Publish/subscribe brokers enable the efficient dissemination of events to a large number of subscribers in support of smart city applications. These events convey data gathered from devices and published to named logical channels called topics. Software-Defined Networking (SDN) can provide the advantage of balancing the load between brokers by switching topics between brokers. However, this switching results in network overhead. Besides, supporting data and decision fusion applications is a challenging task since sensory data has to be fused before being forwarded to subscribers. Therefore, we propose an algorithm utilized by the SDN controller to minimize the load difference between brokers while respecting a reconfiguration limit in support of data and decision fusion applications. We formulate minimizing brokers' load difference within a reconfiguration budget with the constraint of indivisible topics as an Integer Linear Programing (ILP) problem. We show that the problem is NP-Hard and propose a heuristic driven by long-term statistics of topics. The proposed heuristic is evaluated with realistic simulation traffic traces and compared against a threshold-based baseline heuristic driven by instantaneous statistics of topics. Results show that the proposed heuristic performs up to 2000% better load distribution than the baseline heuristic and at least 27% less topic switching.**

***Index Terms*—Brokers, clustering, publish/subscribe, SDN, smart city, topic switching.**

I. Introduction

In 2016, 54.4% of the worlds population lived in cities. This percentage is expected to increase to 60% by 2030 [1]. In addition, 8.4 billion "things" will be connected to the Internet in 2017, increasing to 20.4 billion by 2020 [2]. The trend of moving to urban centers and the increasing number of connected things in the cities require infrastructure and services. These services and infrastructure must meet the needs of citizens and visitors, which open the door for the development of smart cities. To gather data in smart cities, topic-based publish/subscribe protocols let Internet of Things (IoT) devices publish their data as messages to named logical channels, known as topics. Subscribers register for services based on these topics. To forward messages between publishers and subscribers, smart city requires a cluster of brokers in the cloud.

978-1-5386-3531-5/17/$31.00

Software-Defined Networking (SDN) is the technology of choice for managing the network in the context of smart city [3]. The significant benefits of using SDN with IoT are presented in [4]. SDN switches have a flow table with rules that dictate the assignment of topics to brokers. To balance the load between brokers, the SDN controller reconfigures flow table rules in SDN switches (i.e., change assignment of topics to brokers). This reconfiguration routes topic-based messages to the appropriate broker to minimize the load between brokers. Consequently, topics are switched between brokers (i.e., topics shift from one broker to the other). Nevertheless, this switching introduces an overhead on the network.

In this work, we focus on minimizing the load difference between brokers in the cluster given a SDN reconfiguration budget (i.e., topic switching budget) in support of data and decision fusion applications. In data and decision fusion applications, multiple sensory data messages are fused or processed by a broker and results are forwarded to the subscribers in one message. For example, in Body Sensor Networks (BSN), a topic can be the fusion of physiological and psychological sensory data sent to subscribers (e.g., caregivers).

In the case of divisible topics, where subscribers of a topic are distributed across brokers, balancing the load between brokers is a simple problem. However, sensory data messages are forwarded from the SDN switch to all brokers, which introduces an overhead on the network. Besides, same sensory data processing or fusing will be repeated in all brokers.

To provide an efficient performance while supporting data and decision fusion applications, all sensory data messages that share the same topic are routed to the same broker. Therefore, subscribers per topic are not distributed (i.e., topic is indivisible) across multiple brokers. Consequently, network overhead is reduced but balancing the load between brokers is a challenge.

To address this problem, we propose a system that utilizes (i) SDN network with topic-based publish/subscribe

protocol, such as Message Queuing Telemetry Transport (MQTT) or Apache Kafka, and (ii) a long-term topic balancing heuristic on the SDN controller, which is the contribution of this paper. We formulate the problem as an Integer Linear Programming (ILP) problem and show that it is NP-Hard. To evaluate the performance of the proposed heuristic, a threshold-based topic balancing baseline heuristic is implemented. Both heuristics are evaluated using realistic simulation of traffic traces. Results show that the proposed heuristic performs up to 2000% better than the baseline in load difference and at least 27% less topic switching.

II. Related Works

Conventional load balancing algorithms try to strike a balance between brokers using greedy strategies. They do so by distributing the load between brokers to the extent possible every time step; thus, resulting in high topic switching. The literature is rich with research on routing optimization of topic-based publish/subscribe systems [5]. However, using SDN with these systems is first studied in [6].

Wang et al. [5] propose a system named SDNPS, which utilizes SDN central control with topic-based publish/subscribe system for better and non-redundant topic-based events dissemination. SDNPS works by abstracting and aggregating the link state of the network to capture a general overview of the topology. Next, the system predicts traffic distribution for the future. Finally, per-topic minimum overlay network is calculated and shortest path algorithm is used to extract multiple routing paths. The authors of the SDNPS system claim that their system achieves a better trade off between links' global load balancing and per-topic events' minimum cost forwarding.

Bhowmik et al. [7] propose a scalable publish/subscribe SDN-based middleware named PLEROMA that achieves an effective forwarding at line-rate. They state that publish/subscribe middleware can access SDN switches to increase line-rate performance, bandwidth efficiency, and decrease latency when forwarding messages between publishers and subscribers. However, it is challenging to maintain a line-rate performance since publishers and subscribers change their interest dynamically.

Xu et al. [8] use SDN-based fog computing and implement an MQTT broker at the edge switches instead of the cloud due to its proximity to the data source. They show their technique to improve message delivery performance. Nonetheless, edge switches support simple analysis and can not support complicated computations as offered in the cloud. Whereas, Xia et al. [9] propose a method called community-based load balancing, which utilizes interest similarity in community to cluster brokers in a way that enhances the network performance.

None of the mentioned studies deal with the problem of minimizing the load difference between brokers within a reconfiguration limit. Additionally, data and decision fusion applications in the smart city context were not considered in these studies.

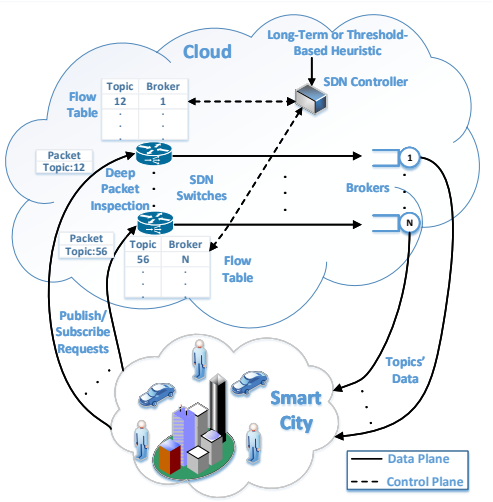


Fig. 1: System model showing the communication between the smart city and the cloud via SDN and publish/subscribe protocols.

III. System Model

We assume a smart city with devices capable of publishing data to the cloud and devices/users willing to subscribe for one or more services in the cloud. We also assume a cloud that consists of an SDN controller with a data flow management algorithm, SDN switches, and N brokers, as shown in Fig. 1. The data flow management algorithm dictates the performance of the system. In this paper, this algorithm is either the proposed or the baseline heuristic. Further, we assume K indivisible topics in the system, each topic i has p_i subscribers.

First, the system run-time period is divided equally into R configuration periods. Additionally, we define the system configuration C during a configuration period as the assignment of topics to brokers as follows:

$$C = S_1, S_2, ..., S_N \tag{1}$$

Where, S_j is the set of topics served by broker j. The system configuration is fixed during a configuration period, which results in zero topic switching. Nonetheless, reconfiguration, which is the transition from one system configuration to another, introduces topic switching.

Second, we define the system load during a configuration period as the load difference between the overloaded and the underloaded brokers in the following equation:

$$L = \max(\{\sum_{i \in S_j} p_i : j = 1 \dots N\}) - \min(\{\sum_{i \in S_j} p_i : j = 1 \dots N\}) \tag{2}$$

Where, L is the system load in number of subscribers, $\min$ and $\max$ are functions that return the minimum and maximum number of subscribers served by any of the brokers respectively.

Publish/subscriber messages sent from the smart city are intercepted by the SDN switches, which in turn uses Deep Packet Inspection (DPI) as a service to extract the topic and the type of message.

For publish messages, the extracted topic is hashed to a number then stored in a predefined field in the incoming message, that flow rules can check. For example, this predefined field can be the Type of Service (TOS) field in the Internet Protocol (IP) header. Overwriting the TOS field does not conflict with the Quality of Service (QoS), since the message is at the penultimate hop before its delivery to the designated broker. For subscribe request messages, a value of zero is injected in a predefined field.

For publish messages, if a flow rule with a matching topic is found, the messages are routed using the broker IP address found in the action field of the flow rule. Otherwise, a `packet-in` message is sent from the SDN switch to the SDN controller. As a result, the heuristic (cf. Section VI) in the SDN controller sends a `packet-out` message for the new topic to the SDN switch with the appropriate broker IP address in the action field.

Subscribe request messages always have a zero value in their predefined field, which is injected by the DPI module. Therefore, the SDN switch sends a `packet-in` message to the SDN controller. The heuristic (cf. section VI) in the SDN controller notices a new subscribe request and extracts the topic number to update the number of subscribers per extracted topic. Next, the heuristic copies the topic number to the predefined field of the message and sends a `packet-out` message to the SDN switch. This `packet-out` message has an action list with IP addresses of all brokers. In other words, for every new subscribe request, the heuristic updates it counters of subscribers per topic and sends a copy of this subscribe request to all brokers. This is important because when the heuristic decides to reassign a topic to a new broker, this new broker knows of all subscribers.

In the next section, we formulate the problem of minimizing the load difference between brokers within a reconfiguration limit as an ILP problem.

IV. Problem Formulation

Finding the best system configuration, that minimizes the load difference between brokers for each of the R configuration periods is a NP-Hard problem. To prove it, we show that the partition problem, which is known to be NP-Hard [10], can be reduced to our problem.

Let $\mathcal{K} = \{p_1, p_2, ..., p_k\}$ be a multiset (set with repeated items) of all topics' sizes. Assuming a cluster of two brokers, we want to partition multiset $\mathcal{K}$ into two subsets $\mathcal{K}_1$ and $\mathcal{K}_2$. In addition, each subset is assigned to a broker, such that the sum of elements in $\mathcal{K}_1$ is equal to the sum of elements in $\mathcal{K}_2$. In other words, the load difference between the two brokers is zero, which is the definition of our problem. However, instead of having two subsets, there are N subsets in our problem.

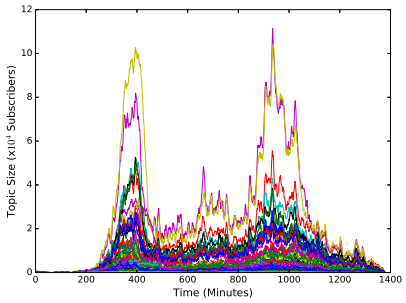
We define a_i^j as a selector with a value of one if topic i is assigned to broker j and zero otherwise. The optimization formulation of the problem is as follows:

$$\text{minimize} \sum_{j=1}^{N}\sum_{q=1}^{N}\left(\left|\sum_{i=1}^{K} a_i^j \cdot p_i - \sum_{i=1}^{K} a_i^q \cdot p_i\right|\right) \quad (3)$$

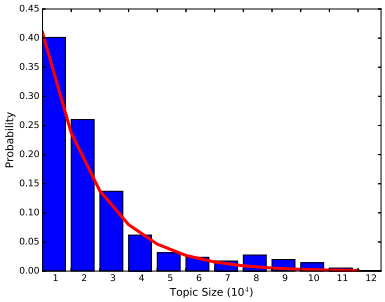
$$s.t. \quad \sum_{j=1}^{N} a_i^j = 1 \qquad \forall i \in 1 \dots K \quad (4)$$

$$a_i^j \in \{0,1\} \qquad \forall i \in 1 \dots K, \forall j \in 1 \dots N \quad (5)$$

Constraint (4) indicates that a topic can not exist in more than one subset while constraint (5) emphasizes that the problem is an ILP problem (i.e., no fractional assignment of topics to subsets).



(a) Number of subscribers per topic over time.



(b) Probability distribution of topics' sizes.

Fig. 2: Dataset characteristics.

V. Dataset

Actual IoT brokers are available and used in the industry. However, their traffic traces are not publicly available. Therefore, we use vehicular traffic with the number of vehicles as a proxy metric for the number of subscribers for smart city applications. In this work, we use the German city of Cologne's realistic simulation scenario that describes the traffic in the city during a 23 hour period. This traffic is based on traveling habits of the inhabitants [11].

The city of Cologne spans a $W \times H$ km^2 area. We divide the city into K zones and each zone is $W_z \times H_z$ km^2. Every zone represents a topic i and the number of vehicles moving within it's boundaries is the number of subscribers p_i. The event of a vehicle entering the boundaries of another zone is interpreted as a new subscribe request for a topic. Therefore, by recording the number of vehicles per zone per time unit, we capture the number of subscribers per topic per time unit.

In our simulation experiments, vehicles' details (e.g., location coordinates) are recorded every second. However, only a few cars can cross the borders of a zone within this period. Consequently, the variance in the number of subscribers per topic is very low, which does not simulate a publish/subscribe system for a smart city. Thus, we recorded the number of vehicles (i.e., subscribers) per zone (i.e., topic) every 60 seconds and noticed that the number of subscribers per topic is around hundreds of thousands.

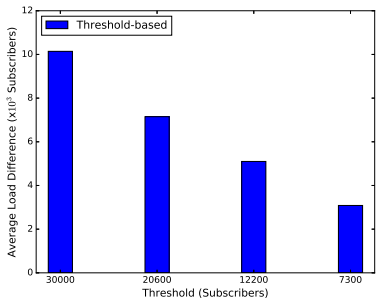
While analyzing the dataset, available in [12], we studied the evolution of topics over time and observed that topics change their size frequently and unpredictably as shown in Fig. 2(a). Additionally, we noticed that topics are much like elephant and mice flows in the Internet traffic [13]. The majority of topics are small in size (mice) while the minority of topics are large in size (elephants) as shown in Fig. 2(b). In fact, we fit the probability distribution of topics based on size as a Pareto distribution. This implies that small-sized topics have a power-law functional relationship with large-sized topics, as illustrated in Fig. 2(b).

In the next section, we develop a long-term algorithm that exploits this observation.

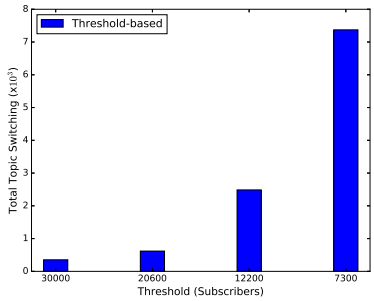
VI. Heuristic Solution

The current problem can be solved using a greedy heuristic that considers instantaneous statistics of topics. It reconfigures SDN switches every time step striving for optimal load distribution between brokers. Nevertheless, the greedy heuristic suffers from very high topic switching.

To provide a balance between the load difference between brokers and the number of reconfigurations, we implement a threshold-based topic balancing heuristic as a baseline heuristic. For every time step, the baseline heuristic computes the current brokers' load difference L_1 using the assignment of topics to brokers used in the last step. Additionally, it sorts topics based on size in descending order then iteratively assigns next topic from the sorted list to the broker with the minimum number of subscribers (i.e., load). Using the new assignment of topic to brokers, L_2 is computed. Finally, the gain in load difference $L_1 - L_2$ is computed. If this gain is less than the threshold then no topic switching is performed. Otherwise, if the gain is greater than the threshold then the system is reconfigured using the new assignment of topics to brokers.



(a) Average load difference v.s. threshold.



(b) Total topic switching v.s. threshold.

Fig. 3: Load difference and topic switching of the baseline heuristic for different threshold values.

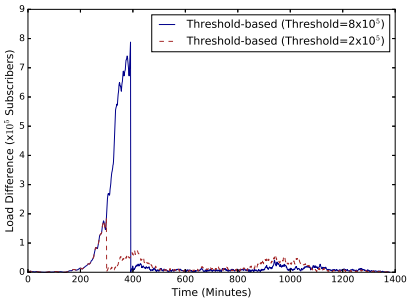


Fig. 4: Load difference of the baseline heuristic over time for two thresholds with both having one reconfiguration but different average load differences.

A low threshold value leads to a better load difference on the expense of more reconfiguration (i.e., high topic switching). While a high threshold value results in a high load difference but less reconfiguration (i.e., less topic switching), as illustrated in Fig. 3. Finding the best threshold is difficult since it is an optimization problem. Furthermore, some threshold values are misleading. For example, Fig. 4 shows that threshold values 200,000 and 800,000 have the same number of reconfigurations. However, the high threshold of 800,000 results in more than two times the load difference obtained using a low threshold of 200,000.

Instead of looking for the best threshold, the proposed heuristic divides the time interval into one or more configuration periods depending on the reconfiguration limit. Next, the heuristic analyzes each configuration

period to come up with a configuration that minimizes the load difference. A limit of one configuration period results in an optimal (i.e., zero) topic switching with good load difference. By increasing the number of configuration periods, topic switching is increased while load difference is enhanced or decreased.

Algorithm 1 presents the proposed heuristic. It performs the following steps for every configuration period in the time range ($t_s \dots t_e$):

- **Line 3:** Find the ideal broker load by distributing the number of subscribers evenly on all brokers to have a perfect load balance using the following:

$$O_t = \frac{\sum_{i=1}^{K} p_i^t}{N} \quad \forall t = t_s \dots t_e \tag{6}$$

- **Line 4:** For every topic, find the number of subscribers in a given sub-interval as shown below:

$$P_i = \sum_{t=t_s}^{t_e} p_i^t \quad \forall i = 1 \dots K \tag{7}$$

- **Line 5:** Sort the set P, set of P_i's, in descending order so that large size topics are fitted first.
- **Lines 6 – 7:** Distribute the first N topics (or less) based on the order of P over brokers (i.e., first topic according to the order of P is assigned to the first broker, second topic to second broker, and so on).
- **Line 8:** For the remaining topics, do the following:
  - Find the total number of subscribers per time step over the configuration period for every broker based on S_j, which is the set of topics assigned so far for broker j as shown below:

$$B_j^t = \sum_{i \in S_j} p_i^t \quad \forall j = 1 \dots N, \forall t = t_s \dots t_e \tag{8}$$

  - Find the difference in the number of subscribers between the ideal broker load (O_t) and broker j for every time step of the configuration period as shown below:

$$D_j^t = O_t - B_j^t \quad \forall j = 1 \dots N, \forall t = t_s \dots t_e \tag{9}$$

  - Find the difference in the number of subscribers between the ideal broker load (O_t) and broker j over the overall configuration period as shown below:

$$D_j = \sum_{t=t_s}^{t_e} D_j^t \quad \forall j = 1 \dots N \tag{10}$$

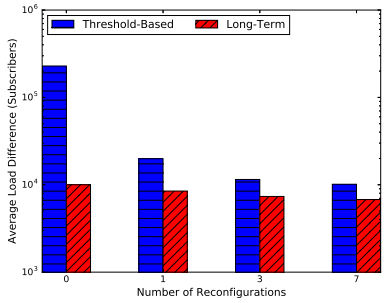
  - Select the broker with the maximum D_j to assign the next topic in P.

VII. Results and Discussions

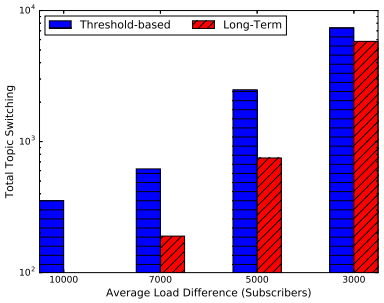
We conducted simulation experiments to evaluate the performance of the proposed heuristic against a baseline heuristic that utilizes a fixed threshold. Performance results indicate that unlike the baseline heuristic, the proposed heuristic provides predictable and improved performance as shown in Fig. 5. The proposed heuristic

Algorithm 1 Long-Term Topic Balancing

1: Input: Number of subscribers per topic over time, p_i^t for time range ($t_s \dots t_e$)
2: Output: System configuration consisting of topics assignment to brokers
3: $\forall t \in \{t_s, \dots, t_e\}$, compute O_t
4: $\forall i \in \{1, \dots, K\}$, compute P_i
5: Sort set P, set of P_i's, in descending order
6: Set M = min(K, N)
7: $\forall j \in \{1, \dots, M\}$, initialize broker j with topic j from set P and compute D_j
8: $\forall j \in \{M+1, \dots, K\}$, assign topic i to the broker with max(D_j) then recompute D_j



(a) Average load difference v.s. reconfiguration.

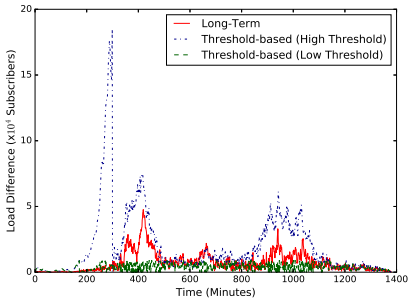


(b) Total topic switching v.s. average load difference.

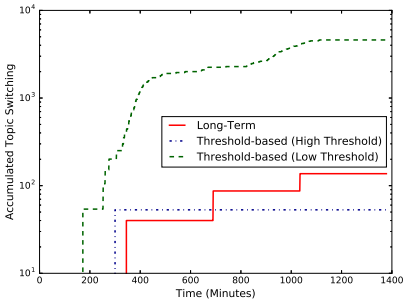
Fig. 5: Average load difference and total topic switching of the proposed and the baseline heuristics.

guarantees an inversely proportional relationship between the number of reconfigurations (i.e., topic switching) and the load difference between brokers. On the other hand, a higher threshold for the baseline heuristic does not guarantee a lower number of reconfiguration as shown in Fig. 4, which leads to unpredictable results.

Moreover, increasing the length of configuration periods (i.e., reducing the number of reconfigurations) widens the performance gap between the two heuristics. For example, when using zero reconfigurations, the proposed heuristic outperforms the baseline heuristic with about 2000% less brokers' load difference, which is clearly seen in Fig. 5(a) and Fig. 7. Also, Fig. 5(b) shows the optimal (i.e., zero) topic switching of the proposed heuristic compared to hundreds of topic switching for



(a) Load difference over time for 3 reconfigurations.



(b) Topic switching over time for 3 reconfigurations.

Fig. 6: Performance of the proposed heuristic using 3 reconfigurations limit and the baseline heuristics using two thresholds.

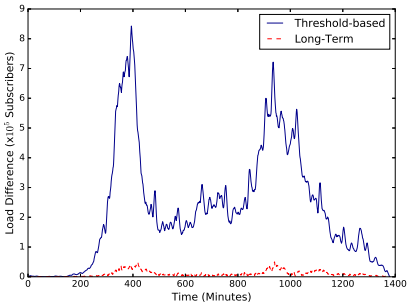


Fig. 7: Brokers' load difference over time of the proposed and the baseline heuristics with zero reconfigurations.

the baseline heuristic.

While proposed heuristic is not optimal, our simulation experiments indicate that it provides better tradeoff between load difference and topic switching, compared to the fixed threshold heuristic. That is, the baseline heuristic can obtain better load difference or topic switching but not both as shown in Fig. 6. Finding the best threshold for the baseline heuristic can cause it to outperform the proposed heuristic with respect to load difference but underperform with respect to topic switching, and vice versa.

VIII. Conclusions and Future Work

In this paper, we address the problem of minimizing the load difference between brokers given a reconfiguration budget in support of data and decision fusion applications in the context of smart cities. We formulate this problem as an ILP problem and show it to be NP-Hard. Due to the complexity of the problem, a heuristic based on the long-term statistics of topics is proposed. The proposed heuristic and a threshold-based baseline heuristic are evaluated using real traffic traces. Results show that the proposed heuristic outperforms the baseline heuristic with more than 2000% better load distribution and at least 27% less topic-switching.

In the future, we plan to evaluate the proposed heuristic via large-scale experiments on a cluster of computers using actual IoT data. We also plan to explore the potential to design an online algorithm that provides worst-case performance guarantees.

IX. Acknowledgment

This publication was made possible by NPRP grant # [71113-1-199] from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

[1] United Nations Population Division, "Data booklet - the world's cities in 2016," 2016. [Online]. Available: http://www.un.org/en/development/desa/population/publications/pdf/urbanization/the_worlds_cities_in_2016_data_booklet.pdf

[2] Gartner- Newsroom, "Gartner says 8.4 billion connected "things" will be in use in 2017, up 31 percent from 2016," 2017. [Online]. Available: http://www.gartner.com/newsroom/id/3598917

[3] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A resilient internet of things architecture for smart cities," *Annals of Telecommunications*, vol. 72, no. 1, pp. 19–30, 2017. [Online]. Available: http://dx.doi.org/10.1007/s12243-016-0530-y

[4] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for internet-of-things: A review," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453–463, Aug 2016.

[5] Y. Wang, Y. Zhang, and J. Chen, "SDNPS: A Load-Balanced Topic-Based Publish/Subscribe System in Software-Defined Networking," *Applied Sciences*, vol. 6, no. 4, p. 91, Mar. 2016. [Online]. Available: http://www.mdpi.com/2076-3417/6/4/91

[6] K. Zhang and H. Jacobsen, "Sdn-like: The next generation of pub/sub," *CoRR*, vol. abs/1308.0056, 2013. [Online]. Available: http://arxiv.org/abs/1308.0056

[7] S. Bhowmik, M. A. Tariq, B. Koldehofe, F. Drr, T. Kohler, and K. Rothermel, "High performance publish/subscribe middleware in software-defined networks," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–16, 2017.

[8] Y. Xu, V. Mahendran, and S. Radhakrishnan, "Towards sdn-based fog computing: Mqtt broker virtualization for effective and reliable delivery," in *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*, Jan 2016, pp. 1–6.

[9] F. Xia, A. M. Ahmed, L. T. Yang, and Z. Luo, "Community-based event dissemination with optimal load balancing," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1857–1869, July 2015.

[10] T. F. Gonzalez, *Handbook of Approximation Algorithms and Metaheuristics (Chapman & Hall/Crc Computer & Information Science Series)*. Chapman & Hall/CRC, 2007.

[11] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, May 2014.

[12] [Online]. Available: https://www.dropbox.com/sh/kdrh3lubaxkrq4c/AACMlzRCGGV_Hk4om5e7qTMha?dl=0

[13] L. Guo and I. Matta, "The war between mice and elephants," in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001*, Nov 2001, pp. 180–188.